IS COTS THE BEST SOLUTION FOR THE SOFTWARE CRISIS IN THE DoD

November 22, 1999

Introduction: In the beginning of software development, we used machine language of ones and zeroes to run our computer programs. We then moved on to procedural languages, which took care of the hands on approach used to design a program. We now have a higher level of abstraction with (COTS) commercial-off-the-shelf software.

The DoD, to try to save money and keep current on the technical advances in the private sector have issued a mandate to challenge system developers to incorporate COTS components into systems without losing any reliability and accessibility of DoD applications. In a letter from the Defense Science Board Task Force which concluded "That DoD's investment in software requires greater DoD-wide management control and oversight in the coming years if the department is to exploit the use of commercial software acquisition practices fully, as well as rapid advances in software technology".[1]

If you look at COTS it seems like a straightforward solution to the software crisis. We in the software industry are always looking for the next best thing in software development and the easiest way to create a software application. But, we need to fully identify the advantages and disadvantages associated with the use of COTS software. There also is a need to have a thorough evaluation of the COTS software package to analyze if these meet the requirements of given software project and overall system performance and is this the most cost effective in the long run for purchase by the DoD.

Advantages: With government budgets getting smaller in size every fiscal year, the emphasis has turned to using COTS products. COTS resources are a wide variety from Microsoft Office Desktop products that are functional with other components. To very complex software

[1] Report of the Defense Science Board Task Force on Acquiring Defense Software Commercially

applications that can be customized to adapt to many different types of architectures.

The main reason the government mandated the use of COTS was to lower maintenance costs. Cost is a very big factor in the classical "make versus buy" question. During a reengineering project, the savings factor can be a tremendous boost to the acquisition of a COTS software product. With COTS-based software a system will be upgraded with new version releases anytime the vendor implements changes. This will keep the technical aspect up to date with the most cutting edge products.

The big push right now is towards COTS products, which takes place in the context of COTS-based systems. The benefits of purchasing a stand-alone Oracle database management system (DBMS) outweigh the cost of creating a functional equivalent from scratch. The government with these commercial systems ultimately wishes that with lower maintenance requirements they would be able to produce significantly lower costs.

By introducing COTS another factor is that systems could have "plug and play" capabilities. "Plug and play" is a big motivation to acquiring COTS software. To which you can have a software environment in which heterogeneous components can be added or changed but also interconnected or interoperated without causing a ripple effect to other related components. This idea was originally used to a very high degree in the hardware venue. The idea was you could change monitors, keyboards etc. from one vendor's product to another without having to load a lot of different types of drivers and change the configuration file.

With client-server systems having the most impact on today's computer system design, and it being still in the early stages of development. COTS products furnish needed supporting technical functionality to change collections of different computing platforms into united, distributed computing environments. The COTS software products, which are available, offer

varying ranges of standards conformity, heterogeneous computing platform support, interoperability, security functionality, performance proficiency, and distributed environment transparency for applications using their services.

Disadvantages: The vendor plays a significant role in whether to purchase a particular COTS product or not. You can become too dependent on a particular vendor's product. You may find some features attractive, but to achieve this will cost more and could make you hostage to a single vendors set of products sometimes called "vendor lock". [2]

You have to investigate if the vendor has a reputable history of providing support at critical times, or if you can get 24-hour seven days a week support for the particular COTS software product you decide to purchase. Will this component be part of the operating system for years to come? Then you hope that the company that produces the software that you choose to integrate will not go out of business, leaving you as a customer with nonfunctional tools and unretrievable data. Even if the vendor is around for years, they often phase out support for the product over a specified time period usually agreed upon during contractual negotiations. You as a customer might have different needs than an ordinary company.

Another point is do they supply well-written, easily understandable documentation for software maintenance? Most of the time the installation documentation is insufficient in providing a good roadmap for the system administrator. There is usually a lot of integration between different interfaces that never gets documented properly leaving you in the dark and guessing if the right parameters are being used turning a system integration into a huge guessing game.

[2] The Commandments of COTS: Still in Search of the Promised Land

When the government purchases a COTS product they are under the illusion of getting cheaper and easier system maintenance. We have to look at the different aspects of how an upgrade can affect a system with a number of Off-the-shelf components. Upgrading a system that has COTS-based software means that as new releases for COTS components are issued by the diverse number of vendors that the system will incorporate all the changes for better or worse. Although a system administrator can choose to skip over certain releases the vendor will tend to support only a limited number of versions. If you elect to skip versions your program will not survive in the long run. But you should keep your systems commercial components as current as possible to save a configuration management headache.

A system with numerous commercial components is very dependent on the various releases of the supporting COTS vendors. The system will require costly licenses to be renewed periodically due to the different components being upgraded at widely varying intervals. A component upgrade, which can result in many unforeseen problems, such as incompatible files and databases, altered naming conventions, and COTS components having new conflicts amongst themselves. The effects of these numerous dependencies of COTS components and different COTS vendors can vary from small amounts of user inconvenience to all out system instability.

In the advantages it was suggested that it would be cheaper to be able to just plug in a software component and have no repercussions. But in the current state of the software world it is quite different than the hardware world and COTS components are rarely built to plug into an existing system without any side effects. The standard ways to overcome these deficiencies are to use “wrappers”, “bridges”, or other “glueware”. The meanings of these terms are third-party software that accomplishes whatever integrating functions needed: taking output from one

component and reformatting it for input to another, sending notification messages about one tool's completion to another for start-up and so forth. The maintenance cost is actually not lower to write the wrappers, which could be a very complex task. Then you have to the knowledge at both the detailed system level and in the COTS components being wrapped. When a new version is released the wrapper or glueware will have the potential of needing to be upgraded. This could be a maintenance nightmare in keeping the glueware up-to-date for any integrated system due to the random vendor release cycles.

The COTS system will still have a critical need to be engineered. You will still have to pay for a system engineer to maintain the software. It will still have its own requirements in the lifecycle process. It will have to have a good design, be coded and integrated, tested independent or otherwise, and managed the same as any other system that has been purchased or engineered in the past.

You cannot decide as just an individual to switch to COTS based systems, it is a very large undertaking which, needs a whole paradigm shift by the entire organization. This is a significant shift from building from scratch to the next of integration of ready-made components. This shift to the new mindset should take place all the way from upper level management through the software developers, software quality assurance, configuration management, and software testing. The managers need to modify their expectations of the techniques used by all these positions because they are going to change with the use of COTS products.

The experience level is very little right now in the government with building and acquiring systems mainly COTS-based over the whole entire lifecycle. The vendors are just giving us promises and wishful thinking instead of hard facts or verified cost models with which to base our excitement for each of their products. We hope to compare the COTS products to

say like the automobile industry, which uses a per-unit component cost consideration. But can we truly put a cost per-unit on software components where each one is different unto itself. There are other hidden costs, which need to be analyzed including:

- "Market research to find COTS products that are suitable."
- "Product analyses to select among alternatives."
- "Licenses and warranties especially if the warranty available to the general public does not suit you're needs."[3]

For each situation the costs that come up consistently to the front depends on each individual circumstance and system, the specific Risk-mitigation plans and the skills of the current management.

Conclusions: We in the software industry are still looking for that magic "Silver bullet", that one thing that will do the next impossible task. I think that the government might be grasping at the COTS products bandwagon as the next one.

The software crisis does tend to exist at different times in the government depending on the out of control software costs and growing system complexities or whichever new conceptual idea someone is trying to push to the top brass. But I believe with proper training and the use of contractor's influx of new and cutting edge concepts we could in the future use the COTS software products to our overall advantage.

Hopefully by doing a very thorough analysis of each product will lead to cutting costs, gaining knowledge and getting cutting edge products used throughout the United States Air Force.

[3] The Commandments of COTS: Still in Search of the Promised Land

Sources:

A Software Development Process for COTS-Based Information System Infrastructure

Part II: Lessons Learned

The Commandments of COTS: Still in Search of the Promised Land

Simplex Architecture: Meeting the Challenges of Using COTS in High-Reliability Systems

The Opportunities and Complexities of Applying Commercial-Off-the-Shelf Components

A Software Development Process for COTS-Based Information System Infrastructure: Part 1

Report of the Defense Science Board Task Force on Acquiring Defense Software Commercially (Office of the Under Secretary of Defense for Acquisition & Technology)

No Silver Bullet Essence and Accidents of Software Engineering